



JFR - JDK Flight Recorder

JFR - JDK Flight Recorder

Overview

Definition from docs.oracle.com :

Flight Recorder (JFR) is a profiling and event collection framework built into the JDK.

Flight Recorder allows Java administrators and developers to gather detailed low-level information about how a JVM and Java applications are behaving.

How does it work

Collecting data & overhead

JFR collects diagnostic and profiling data about a running Java application and the JVM it is running on.

Because it is deeply embedded into the JVM, the overhead is typically very small : **Oracle** claims a drop of less than 1% in performance while using JFR in heavy production environments.

Overhead might be higher in short-lived applications (because of startup and warmup time) such as microservices but this is not the case of AeroWebb.

To achieve this low overhead, JFR collects data as events. From docs.oracle.com :

Events occur in the JVM or the Java application at a specific point in time. Each event has a name, a time stamp, and an optional payload.

The payload is the data associated with an event, for example, the CPU usage, the Java heap size before and after the event, the thread ID of the lock holder, and so on.

Because JFR is sitting at the core of the JVM, it can record events at an extremely fine level. To further limit overhead and keep the amount of collected data down, you can selectively choose the types of events you want to monitor.

Writing data : .jfr files

Event data is stored in buffers at the thread-local level before being merged into a global in-memory buffer and finally written to disk.

Any data written to disk is flushed from the memory, so there is no overlap of information between the two.

You can choose to disable automatic disk operations and only write to disk on demand : this is especially useful to keep overhead down in production environments and using JFR as diagnostic tool that only reports events as issue arise.

Data is written in the form of a binary file (.jfr) containing all events (and their metadata) that occurred during a recording.

How to use it : recording data

Enabling JFR

JFR is disabled by default. To enable JFR in JDK11, we need to launch the Java application we want to monitor with the `-XX:+FlightRecorder` option.

 Configuration for AeroWebb in `/tomcat/bin/setenv.bat`

```
set CATALINA_OPTS=-XX:+FlightRecorder
```

Control recordings

Recordings can be started on application startup or on demand. They can be of a fixed or continuous duration.

You can start/stop recordings in 3 different ways :

On application start - command line

You can pass the `-XX:StartFlightRecording` option to your Java application to start a recording on startup.

See [the documentation](#) for more information on usable parameters.

 Configuration for AeroWebb in `/tomcat/bin/setenv.bat`

```
set CATALINA_OPTS=-XX:+FlightRecorder -XX:StartFlightRecording=duration=10s,filename=onStartup.jfr
```

This will start a recording on application startup that will collect events for 10s and write all information in a file name `onStartup.jfr`.

Using jcmd

`jcmd` is a tool bundled with the JDK (in the `jdk/bin` folder) that can be used to send diagnostic commands to a running JVM. To see a list of available commands, you can type :

```
jcmd <app_pid>
```

where `<app_pid>` is the pid of a running Java application. This information is also available in the documentation.

Tip

On Windows, the following command can help you to quickly find the pid of the process listening on port 8080 (Tomcat with AeroWebb in a default local environment, customize for your needs) :

```
netstat -aon | find /i "listening" | find "8080"
```

Usage of jcmd

```
jcmd <PID> JFR.start duration=10s filename=withJCMD.jfr
```

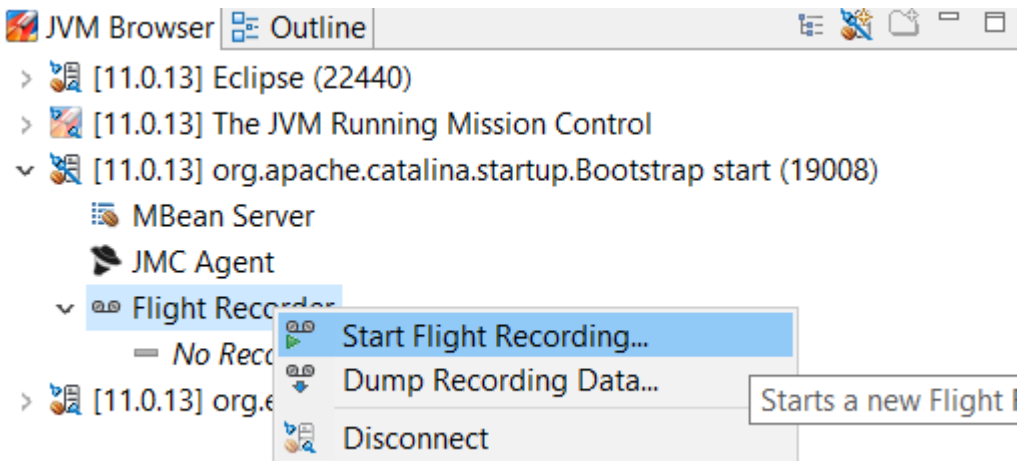
This will start a recording for the process with id <PID> that will collect events for 10s and write all information in a file name `withJCMD.jfr`.

Using JMC

Please refer to the [JMC page](#) for more information.

In JMC, find your application in the JVM browser and right click on the Flight Recorder (see figure 1) to open a wizard (see figure 2) guiding you through configuration of a recording.

JVM Browser



Wizard

Start Flight Recording

Start Flight Recording

Edit recording settings and then click Finish to start the flight recording.

Destination File:

Name:

Time fixed recording

Recording time:

Continuous recording

Maximum size:

Maximum age:

Event settings:

Description:

Note: Time fixed recordings will be automatically dumped and opened.